REMARKS

This Amendment is filed in response to the Office Action dated July 14, 2008. All objections and rejections are respectfully traversed.

Claims 1-72 are in the case.

Request for Interview

The Applicant respectfully requests a telephone interview with the Examiner after the Examiner has had an opportunity to consider this Amendment, but before the issuance of the next Office Action. The Applicant may be reached at 617-951-2500.

Rejection Under 35 U.S.C. §112

At paragraph 4 of the Office Action, claims 1-21, 28-33, and 71-72 were rejected under 35 U.S.C §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter to which applicant regards as the invention. Applicant believes amended claims meet the requirements of 35 U.S.C § 112, second paragraph and are in condition for allowance.

Rejection Under 35 U.S.C. §102(b)

At paragraph 7 of the Office Action, claims 1-14 and 16-72 were rejected under 35 U.S.C. §102(b) as being unpatentable in view of Permut U.S. Patent No. 6,260,115, issued on July 10, 2001 (hereinafter "Permut").

Applicant's claimed invention, as set forth in representative claim 1, comprises in part:

> 1. A method for a storage operating system implemented in a storage system to optimize an amount of readahead data retrieved from a data container of the storage system, the method comprising:
>
> maintaining a plurality of files;
>
> maintaining, for a selected file of the plurality of files, a plurality of readset data structures, each readset data structure holding a plurality of factors for a selected readstream, ***the plurality of factors allowing the system to adjust adaptively the amount of data retrieved from the data container***;
>
> receiving a client read request for a particular read stream at the storage system;
>
> locating a readset data structure for the particular read stream;
>
> determining whether the storage operating system is permitted to retrieve readahead data from the data container in response to the received client read request;
>
> if it is determined that the storage operating system is permitted to retrieve readahead data from the data container, performing the steps of ,
>
> (i) ***adjusting an amount of readahead data to retrieve from the data container, based on the plurality of factors stored within the readset data structure;***
>
> (ii) retrieving the adjusted amount of readahead data from the data container; and
>
> (iii) determining if the readahead data structure meets a criteria for being updated, and if the readahead data structure meets the criteria then updating the readahead data structure.

Permut discloses a method for detecting and remembering sequential access patterns for the purpose of prestaging tracks ahead of the current access request. Specifically, the number of tracks requested to be prestaged ahead may be responsive to the amount of storage available in the cache memory. In his Abstract Permut states:

> "Once a sequential access pattern is detected, one or more tracks are requested to be prestaged ahead of the current access request. The number of

tracks requested to be prestaged ahead may be responsive to the amount of storage available in the cache memory."

This tells the reader that Permut uses a previous access to determine how much prestage data should be collected from the data container. Permut does not allow the system to adjust based on a plurality of factors, but rather only based on prior accesses.

In his Fig. 7A, Fig. 7B, Fig. 7C, and Fig. 7D, Permut determines whether the prestaged tracks are sequential, and if prestaging is done in response to included hints. Hints are defined as alerts, indicating that the application is likely to access data sequentially for some number of tracks or blocks. Therefore, a hint determines whether or not a fixed number of data blocks should be prestaged. Permut does not adjust an amount of readahead data to be retrieved from the data container, based on a plurality of factors stored within a readset data structure.

Applicant respectfully urges that Permut has no disclosure of Applicant's claimed novel ***adjusting an amount of readahead data to retrieve from the data container, based on the plurality of factors stored within the readset data structure.***

Applicant's claimed invention is a method and system for a storage operating system, implemented in a storage system, to optimize an amount of readahead data retrieved from a data container of the storage system. In particular, a plurality of readset data structures are **maintained for a selected file of the plurality of files**. Each of **the readset data structures hold a plurality of factors for a selected readstream that allow the system to adaptively adjust the readahead data** accordingly. The system initiates when a client read request is **received at the storage system for a particular read request**. Once the request has been received, the system locates a readset data structure for the particular read stream so that the system can determine whether the storage operating system is permitted to retrieve readahead data from the data container in response to the received client read request. If it is determined that the storage operating system is permitted to retrieve readahead data from the data container, the system acts accordingly and (i) ***adjust the amount of readahead data to retrieve from the data container. This adjustment is based on the plurality of factors stored within the readset data structure, (ii) retrieves the adjusted amount of readahead data from the data container, and (iii) determines if the readset data structure meets a criteria for being updated, and if the readset data structure meets the criteria, updating the readset data structure.***

Permut does not disclose adjusting the amount of readahead data to retrieve from the data container based on a plurality of factors. Permut is merely a flag based system which indicates whether the system should or should not retrieve a predetermined, fixed amount of data using Permut's prestaging process. This type of system is like that of the prior art discussed on page 5 of the Applicant's background. Furthermore, as discussed above, Permut does not use a plurality of factors to adjust the amount of data that is retrieved from the data container. Permut assigns a predetermined fixed amount (e.g., hints) connected to file for the system to use the next time the data is retrieved. In Applicant's retrieval process, the amount of data collected during the readahead step is determined by the system each time there is a new read request. **This amount is determined based on a plurality of current system factors not on a previously established fixed rate like in Permut.**

Accordingly, Applicant respectfully urges that Permut is legally incapable of anticipating Applicant's claimed invention under 35 U.S.C. 102 because of the absence from the disclosure of Permut of Applicant's claimed novel ***adjusting an amount of readahead data to retrieve from the data container, based on the plurality of current factors stored within the readset data structure.***

Applicant's Response to Examiner's Response to Arguments

Applicant respectfully points out there is no mention of "***adjusting*** an amount of readahead data ***to retrieve from the data container,*** based on the plurality of factors stored within the readset data structure.". Permut tells the system to retrieve a fixed amount of previously chosen data every time the system retrieves data for a particular file. Permut does not adjust the data to be retrieved every time the data is retrieved. By definition, ***adjust*** means changing. Therefore, a fixed amount can not be changed (i.e., adjusted). The definition provided above is supported in Applicant's Specification on page 21, Lines 16-23, which state:

> "The level value 604 may be ***adjusted …*** after every client read request is processed by the file system 260, until the readset 600 is deallocated or reused, as described below. This aging process may be subject to various conditions. For example, if the level value 604 is decremented to a value that is less than the predetermined initial value (e.g., equal to 10), then the next time the level value is incremented, the level value may be set equal to the predetermined initial level value. Further, the file system 260 may ensure that the level value 604 is not incremented past its predetermined upper-bound value nor decremented below its predetermined lower-bound value."

Further page 29, Lines 2–10 state

> "Specifically, the total number of blocks to read 714 is ***adjusted*** to include both the client-requested must-read data blocks 716 as well as the number of speculative readahead data blocks 718 determined by the steps 920-945. Here, it is noted that the file system 260 searches the buffer cache 156 to determine whether any of the requested data blocks are already stored in in-core buffers 1200. If so, the file system appropriately adjusts the disk I/O hint 710 to retrieve only those data blocks not already accessible in the buffer cache. Moreover, the file system may have to generate more than one disk I/O hint 710 if more than one non-contiguous sequence of file data blocks (i.e., fbns) are requested from disk storage."

Rejection under §103(a)

At paragraph 31 of the Office Action, claim 15 was rejected under 35 U.S.C. §103(a) as being unpatentable over Permut in view of Vishlitzky et al., U.S. Patent No. 5,649,156, issued on July 15, 1997 (hereinafter "Vishlitzky").

Claim 15 is believed to be dependent from allowable independent claim 1, and therefore in condition for allowance.

Conclusion

All independent claims are believed to be in condition for allowance.

All dependent claims are dependent from independent claims which are believed to be in condition for allowance. Accordingly, all dependent claims are believed to be in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

/Stephen D. LeBarron/
Stephen D. LeBarron
Reg. No. 62,479
CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210
Telephone: (617) 951-2500
Facsimile: (617) 951-3927